Amendments to the Claims:

This listing of claims replaces all prior versions and listings of claims in the application:

Listing of Claims:

Claims 1-22. (Canceled)

23. (Currently amended) A method comprising:

in a parallel processor comprising a controlling processor linked to a remote console system and a plurality of micro engines, each of the micro engines comprising a plurality of executable threads, determining if one of the plurality of threads ~~treads~~ is executing in a target micro engine;

in response to determining, pausing execution of the threads;

loading a series of hop instructions from a debug library;

storing program counters for the threads;

modifying the program counters of the target micro engine to jump to a start of the series of hop instructions; and

modifying the series of hop instructions to return to the stored program counters; and

copying the series of hop instructions to an unused segment of micro store in the target micro engine;

executing the series of hop instructions.

24. (Previously presented) The method of claim 23 further comprising resuming execution of the threads in the target micro engine using the stored program counters.

25. (Previously presented) The method of claim 23 wherein pausing comprises receive a pause command from the remote console system.

26. (Currently amended) The method of claim 23 wherein the series of hop instructions are selected by a user of the remote console system.

27. (Currently amended) The method of claim 23 wherein the series of hop instructions control a start and a stop of selected bus ports with each hop.

28. (Currently amended) The method of claim 23 further comprising:

modifying the program counters of the plurality of micro engines to jump to a start of the series of hop instructions; and

modifying the series of hop instructions to return to the stored program counters; and

copying the series of hop instructions to an unused segment of micro store in the micro engines;

executing the series of hop instructions in the micro engines in unison.

29. (Previously presented) The method of claim 28 further comprising resuming execution of the threads in the plurality of micro engines using the stored program counters.

30. (Currently amended) A processor that can execute multiple contexts comprising and that comprises:

a register stack;

a program counter for each executing context;

an arithmetic logic unit coupled to the register stack and a program control store that stores a command that causes the processor to:

determine if one of the multiple contexts is executing in a target micro engine;

in response to determining, pause execution of the multiple contexts;

load a series of hop instructions from a debug library;

store program counters for the multiple contexts;

modify the program counters of the target micro engine to jump to a start of the series of hop instructions; and

modify the series of hop instructions to return to the stored program counters; and

copy the series of hop instructions to an unused segment of a micro store in the target micro engine;

execute the series of hop instructions.

31. (Previously presented) The processor of claim 30 wherein the command further causes the processor to:

resume execution of the contexts in the target micro engine using the stored program counters.

32. (Previously presented) The processor of claim 30 wherein pausing comprises receiving a pause command from a remote console system.

33. (Currently amended) The processor of claim 32 wherein the series of hop instructions are selected by a user of the remote console system.

34. (Currently amended) The processor of claim 30 wherein the series of hop instructions control a start and a stop of selected bus ports with each hop.

35. (Currently amended) The processor of claim 30 wherein the command further causes the processor to:

modify the program counters of the plurality of micro engines to jump to a start of the series of hop instructions;

modify the series of hop instructions to return to the stored program counters; and

copy the series of hop instructions to an unused segment of micro store in the micro engines;

execute the series of hop instructions in the micro engines in unison.

36. (Previously presented) The processor of claim 35 wherein the command further causes the processor to: resume execution of the threads in the plurality of micro engines using the stored program counters.